

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

«До захисту допущено»

В.о. завідувача кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“ ” _____ 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки _____ 6.050103 «Програмна інженерія»

спеціальність _____ «Програмне забезпечення систем»

на тему: _____ WEB-застосування проведення тематичних дискусій та online-опитувань

Виконав: студент 4 курсу, групи ІП-52

_____ Вашенко Юрій Олександрович

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник

_____ ас. каф. АСОІУ Недашківський Євген Анатолійович

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

**Консультант з
графічної
документації**

_____ доц. к.т.н. Ліщук К.І.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент

_____ доц. каф. ОТ, к.т.н., доц. Болдак А.О.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цьому
дипломному проекті немає
запозичень з праць інших
авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України
“Київський політехнічний інститут ім. І.Сікорського”**

Факультет (інститут) _____ Інформатики та обчислювальної техніки
(повна назва)

Кафедра _____ автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки _____ 6.050103 «Програмна інженерія»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“ _____ ” _____ 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Ващенко Юрій Олександрович

(прізвище, ім'я, по батькові)

1. Тема проекту WEB-застосування проведення тематичних дискусій та online-опитувань

керівник проекту Недашківський Євген Анатолійович, ас. кафедри АСОІУ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від « _____ » _____ 2019 р. № _____

2. Термін подання студентом проекту « _____ » _____ 2019 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення

2) Моделювання та конструювання програмного забезпечення

3) Аналіз якості тестування та програмного забезпечення

4) Впровадження та супровід програмного забезпечення

5. Перелік графічного матеріалу (із зазначенням обов'язкових кресленників, плакатів, презентацій тощо)

1) Схема структурна варіантів використання програмного забезпечення

2) Схема структурна баз даних програмного забезпечення

3) Креслення екранних форм

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	завдання видав	завдання прийняв

7. Дата видачі завдання « » 2019 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
	<i>Вивчення предметної області</i>		
2			
	<i>Постановка та формалізація задачі</i>		
4	<i>Аналіз вимог до програмного забезпечення</i>		
5			
	<i>Подання ДП на основний захист</i>		

Студент

(підпис)

Керівник проекту

(підпис)

Ващенко Ю.О.

Недашківський Є.А.

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 6.050103
«Програмна інженерія» (Програмне забезпечення систем)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
О.А. Павлов
(підпис) (ініціали, прізвище)
“ ” 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Ващенко Юрію Олександровичу
(прізвище, ім'я, по батькові)

**1. Тема проекту « WEB-застосування проведення тематичних
дискусій та on-line»**

керівник проекту Крамар Юлія Михайлівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “23” квітня 2019 р. №1181-с

2. Термін подання студентом проекту «03» червня 2019 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

*1) Аналіз вимог до програмного забезпечення: основні визначення та терміни,
опис предметного середовища, огляд існуючих технічних рішень та відомих
програмних продуктів, розробка функціональних та нефункціональних вимог*

*2) Моделювання та конструювання програмного забезпечення: моделювання та
аналіз програмного забезпечення, засоби розробки, технічні рішення,
архітектура
програмного забезпечення*

3) Розгортання та впровадження програмного забезпечення

4) Керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1) *Схема структурна варіантів використань*

2) *Схема бази даних*

3) *Креслення вигляду екранних форм*

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання *«12» березня 2018 року*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>19.03.2019</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>26.03.2019</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>26.03.2019</i>	
4.	<i>Аналіз вимог до програмного забезпечення</i>	<i>02.04.2019</i>	
5.	<i>Алгоритмізація задачі</i>	<i>02.04.2019</i>	
6.	<i>Моделювання програмного забезпечення</i>	<i>09.04.2019</i>	
7.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>16.04.2019</i>	
8.	<i>Розробка архітектури програмного забезпечення</i>	<i>23.04.2019</i>	
9.	<i>Розробка програмного забезпечення</i>	<i>30.04.2019</i>	
10.	<i>Налагодження програми</i>	<i>07.05.2019</i>	
11.	<i>Виконання графічних документів</i>	<i>14.05.2019</i>	
12.	<i>Оформлення пояснювальної записки</i>	<i>21.05.2019</i>	
13.	<i>Подання ДП на попередній захист</i>	<i>30.05.2019</i>	
14.	<i>Подання ДП рецензенту</i>		
15.	<i>Подання ДП на основний захист</i>		

Студент

_____ Ващенко Ю.О.
(підпис)

Керівник проекту

_____ Крамар Ю.М.
(підпис)

[illegible]

Пояснювальна записка до дипломного проекту

на тему: _____

Київ – 2019 року

АНОТАЦІЯ

Пояснювальна записка дипломного проекту складається з чотирьох розділів, містить 35 таблиць, 1 додаток та 12 джерел – загалом 61 сторінок.

Об'єкт дослідження: веб системи моделі клієнт-сервер з реляційними базами даних, що використовуються для роботи сайту для комунікації користувачів.

Мета дипломного проекту: створити веб сервіс, що може бути використаний для проведення дискусійних обговорень та опитувань онлайн.

У першому розділі було проаналізовано предметну модель, виділено функціональні та нефункціональні вимоги, зроблено порівняння за аналогічними проектами.

У другому розділі було розроблено схему бізнес логіки додатку, архітектуру, сконструйовано дипломний проект та проаналізовано його безпеку.

У третьому розділі проведено тестування веб застосунку за розробленим планом тестування. Описано процес тестування.

У четвертому розділі описано розгортання та впровадження веб застосунку, а також наведено схему структурну розгортання.

У додатках наведено: опис програми, схема варіантів використання, схема бази даних та зроблено креслення вигляду екранних форм.

КЛЮЧОВІ СЛОВА: ОНЛАЙН ДИСКУСІЇ, ОНЛАЙН ОПИТУВАННЯ, КОМУНІКАЦІЙНИЙ СЕРВІС

ABSTRACT

The explanatory note of the diploma project consists of four sections, contains 35 tables, 1 application and 12 sources - a total of 61 pages.

Object of research: Web-based client-server model with relational databases used to operate the site for user communication.

The purpose of the diploma project: to create a web service that can be used for conducting discussion discussions and online surveys.

In the first section the subject model was analyzed, functional and non-functional requirements were highlighted, comparisons were made on similar projects.

In the second section the scheme of business logic of the application, the architecture, the diploma project was designed and analyzed its security.

The third section tests the web application for the developed test plan. The testing process is described.

The fourth section describes the deployment and implementation of the web application, as well as the structured deployment scheme.

The annexes provide: a description of the program, a schematic of the use options, a schema of the database and a drawing of the form of the screen forms.

KEYWORDS: ONLINE DISCUSSIONS, ONLINE EXAMINATION, COMMUNICATION SERVICE

					КПІ.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП.....	11
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	12
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	12
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ	13
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	14
Побудуємо РЕЗУЛЬТУЮЧУ МАТРИЦЮ ТРАСУВАННЯ НА РИСУНКУ	27
1.5 РОЗРОБКА НЕФУНКЦІОНАЛЬНИХ ВИМОГ	27
1.6 ПОСТАНОВКА ЗАДАЧІ	27
1.7 ВИСНОВКИ ПО РОЗДІЛУ	28
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 30	
2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	30
2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	36
2.3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	37
2.4 АНАЛІЗ БЕЗПЕКИ ДАНИХ.....	49
2.5 ВИСНОВКИ ПО РОЗДІЛУ	49
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 50	
3.4 АНАЛІЗ ЯКОСТІ ПЗ.....	50
3.5 Підходи до тестування	51
3.5.3 Компонентне тестування.....	51
3.5.4 Інтеграційне тестування.....	52
3.5.5 Тестування продуктивності.....	52
3.6 КРИТЕРІЇ ПРОХОДЖЕННЯ ТЕСТУВАННЯ	52
3.6.3 Компонентне тестування.....	52
3.6.4 Інтеграційне тестування.....	52
3.6.5 Тестування швидкодії.....	53
3.7 ПРОЦЕС ТЕСТУВАННЯ.....	53
3.7.3 Дані до тестів.....	53

3.7.4	Задачі тесту.....	53
3.7.5	План виконання.....	54
3.8	ВИМОГИ ДО СЕРЕДОВИЩА.....	54
3.8.3	Апаратна частина.....	54
3.8.4	Програмна частина.....	54
3.8.5	Вимоги до безпеки.....	54
3.8.6	Інструменти.....	54
3.9	ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ.....	54
4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	57
4.4	РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	57
4.4.3	Встановлення основного сервісу.....	57
4.4.4	Встановити СУБД PostgreSQL.....	57
4.4.5	Налаштувати файл WebApp.dll.config.....	57
4.4.6	Здійснити міграцію структури БД в СУБД.....	58
4.4.7	Запустити основний сервіс.....	58
4.5	ІНСТРУКЦІЯ КОРИСТУВАЧА.....	58
	ВИСНОВКИ	59
	ПЕРЕЛІК ПОСИЛАНЬ.....	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Дискусія – це форма колективного обговорення, мета якого — виявити істину або знайти правильний розв'язок порушеного питання через висловлення власних міркувань та зіставлення поглядів опонентів на проблему.

Пост – це інформаційне повідомлення в мережі, що містить певну мету розміщення, одним із видів посту є стаття.

Фреймворк - бібліотека сирцевого коду, що надає функціонал, ідіоматику та визначає архітектуру додатку при його використанні.

Драйвер – програмний продукт для взаємодії з іншою апаратно-програмною системою.

СУБД – система управління базою даних – програмний продукт для керування сховищем бази даних.

Роути – це серверні налаштування для надання відповідності функціоналу до шляху запиту в URI.

Модуль – структурна одиниця програмного забезпечення, що містить логічно зв'язаний функціонал.

ВСТУП

Інформаційні технології є важливою складовою у сучасному суспільстві. Вони зайняли багато ніш, де є корисними для людини, зокрема дозволяють поширювати знання, обмінюватись думками та емоціями в реальному часі. Це все можливо завдяки мережі інтернет, що має велике поширення та доступна майже всім.

В всесвітній мережі існує безліч сайтів та сервісів, що мають різні функції та цілі, однією з таких цілей є комунікація між людьми. Для того, щоб кожен міг в зручному форматі обмінятись думками з іншими створена велика кількість форумів, чатів, систем ведення блогу. Кожна з цих систем має свою модель яка зручна для якогось із аспектів комунікації. Також сайти можуть мати свою тематику, та додаткові можливості. Були створені різні моделі обміну інформацією, деякі сервіси комбінують їх, це, наприклад, соціальні мережі, інші ж залишаються в рамках однієї. Практика створення рішень для комунікації між людьми досить поширена тому існує багато різноманітних прикладів впровадження цих комунікаційних моделей.

Сучасні рішення для проведення дискусій в веб просторі мають багато цікавих рішень, що робить їх корисними для широкого кола людей та є необхідними в рамках будування спільнот навколо професійних та соціальних тематик.

Мета розробки – систематизування теоретичних знань отриманих впродовж учбового процесу для створення сервісу з можливістю комунікації on-line. Призначення розробки - використання проекту у сфері онлайн сервісів для проведення дискусій.

Завданням даної роботи є розробка ВЕБ-застосунку для проведення онлайн дискусій між користувачами, а також проведенням онлайн опитувань. Компонентами розроблюваної системи є серверний застосунок та база даних. Результатом роботи є система яку можна використовувати в якості сервісу проведення дискусій та опитувань різної тематики.

					КП.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Дискусійний формат обговорень посідає важливе місце в області систем комунікації, він зазвичай має за собою деяку мету – вирішення питання, поширення та якісний аналіз

1.2 Змістовний опис і аналіз предметної області

Дискусійний формат обговорень – це формат що має за мету отримати структуровану та компетентну відповідь на поставлене питання об дати якісний аналіз щодо певного явища. Він має складатись щонайменше з постановки питання або інформаційного повідомлення від автора та обговорення представлених даних від користувачів, зацікавлених в темі. Для цього цей формат повідомлення зазвичай включає такі елементи як пост (інформаційне повідомлення автора), коментарі - обговорення інших користувачів даної теми, до того ж кожен коментар має вказувати або на пост, або на коментар іншого автора, до якого був доданий коментар. Також модель обговорення має включати в себе оцінювання матеріалів лайком або дизлайком.

Така система дає можливість отримати спільноту, що може саморегулюватись шляхом оцінки кожного повідомлення, оскільки це змушує автора робити більш корисний інформаційний внесок в повідомлення. Також це дає можливість виділяти найбільш якісні повідомлення на передній план, тому даний спосіб регуляції є найкращим вибором, коли є необхідність вивчати цей матеріал пізніше. Система побудови коментарів в ієрархічному вигляді дає можливість орієнтуватись в коментарях та вибирати цікавий напрямок обговорення читачу.

Опитування є корисним доповненням до дискусійного формату, оскільки одним з видів обговорення та отримання якісної оцінки будь якому явищу є

					КП.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

вибір з декількох варіантів. В даному проекті буде реалізовано опитування саме у цьому вигляді, тому складовими опитування будемо:

- поставлене питання;
- набір варіантів вибору;
- його результуюча оцінка спільнотою.

Також одним із складових моделі дискусій є поділення на спільноти – групи людей зі спільними інтересами або об'єднаними будь якою темою, що може мати необхідність в обговоренні. Цей поділ є необхідним, тому що зосереджує увагу та змістовний склад обговорень в потрібне русло та не заважає людям не зацікавленим в дискусійній темі.

1.3 Аналіз успішних ІТ-проектів

Серед сервісів зі схожими властивостями можна виділити такі сервіси як Reddit, Nabrahabr, Dou.ua – їх досить багато, оскільки такий формат є зручним для комунікації, коли необхідно дати можливість спілкуватись та мати корисний контент, що зручний читачам. Проте між ними є досить великі відмінності, вони можуть мати додаткові специфічні можливості, користі для загальної тематики сайту, різний користувацький інтерфейс.

Reddit – відомий сайт, що містить загальну тематику, в ньому є розділення на сабреддіти (тематики), в яких можливо створювати пости, коментувати їх, оцінювати як пости так і коментарі. Відображує найкращі відповіді в пріоритеті. Має досить старий традиційний дизайн. В порівнянні з даним проектом Reddit має старий дизайн, не має функціонал швидкого опитування.

Nabrahabr – сайт, що має загальну професійну тематику – ІТ. Містить багато цікавих та професійних статей, головною особливістю є те, що має строгу систему, завдяки якій на ньому коментує та створює пости відносно невелика кількість користувачів, але при цьому дані що публікуються загалом мають більшу інформаційну цінність. Має також рейтингову систему

коментарів та постів, хаби (тематики), блоги компаній. На відміну від даного проекту Nabrahabr має обмежену тематику та рейтинг, що розповсюджується на весь сайт, замість одної спільноти.

Doou.ua – сайт-форум для розробників з України, має менш строгу систему, серед тематики як ІТ-технології, так і працевлаштування в ІТ, має додаткові розділи пов'язані з цим а також містить рейтинг компаній, виділяє найбільш оцінені коментарі. В порівнянні з Doou.ua, даний проект не має спеціалізації на темі та не має специфічного функціоналу – весь контент створюється в рамках моделі спільнот.

1.4 Аналіз вимог до програмного забезпечення

Варіанти використання в системі представлені нижче:

Таблиця 1.1 – Варіант використання UC001

Назва	Авторизація користувача
Опис	Користувач має можливість авторизуватись в системі.
Учасники	Користувач
Передумови	
Постумови	Користувач проходить авторизацію
Основний сценарій	Система показує вікно входу, що містить логін та пароль. Користувач заповнює поля та натискає кнопку «Увійти». Система авторизує користувача
Розширення сценаріїв	4. Система виявляє, що введені дані некоректні 4.1.а Система демонструє повідомлення про помилку.

Таблиця 1.2 – Варіант використання UC002

Назва	Реєстрація користувача
Опис	Користувач має можливість реєструватись в системі.
Учасники	Користувач
Передумови	Користувач не має облікового запис в системі
Постумови	Користувач має обліковий запис в системі
Основний сценарій	Система показує вікно реєстрації з полями «Ім'я», «Стать», «Логін», «Пароль» та кнопку «Реєстрація». Користувач заповнює всі поля, натискає на кнопку та переходить на сторінку користувача.
Розширення сценаріїв	4.1 Система виявляє, що введені логін існує 4.1.a Система демонструє повідомлення про існуючий логін.

Таблиця 1.3 – Варіант використання UC003

Назва	Створення нової спільноти користувачем
Опис	Користувач може створювати спільноти.
Учасники	Користувач
Передумови	Користувач зареєстрований в системі
Постумови	Користувачем створена спільнота
Основний сценарій	Користувач вводить значення в поля «URL спільноти», «Назва спільноти», «Опис спільноти» дані. Та натискає кнопку «Створити спільноту»

Продовження таблиці 1.3

Розширення сценаріїв	<p>4.1 Система виявляє, що введена назва спільноти та URL існує</p> <p>4.1.a Система демонструє повідомлення про існуючу спільноту.</p>
----------------------	---

Таблиця 1.4 – Варіант використання UC004

Назва	Створення посту
Опис	Користувач може створювати пости
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та належить хоча б одній спільноті
Постумови	Створений новий пост користувачем
Основний сценарій	Користувач в вікні створення посту вводить дані в поля «Заголовок посту», «Пост», «Спільнота посту», при необхідності додає опитування, ввівши його назву та вибравши пункти опитування. Натискає кнопку «Створити пост». В системі створюється пост.
Розширення сценаріїв	<p>4.1 Система виявляє, що пост занадто короткий</p> <p>4.1.a Система демонструє повідомлення про мінімальну довжину посту.</p> <p>4.2 Система виявляє, що пост занадто довгий</p> <p>4.2.a Система демонструє повідомлення про максимальну довжину посту.</p>

Таблиця 1.5 – Варіант використання UC005

Назва	Коментування посту, коментарю
Опис	Користувач може коментувати пости та коментарі
Учасники	Користувач, автор посту або коментарю
Передумови	Користувач має обліковий запис в системі та належить спільноті, до якої доданий пост
Постумови	До посту доданий коментар
Основний сценарій	Користувач натискає кнопку «Відповісти» до посту або коментарю. Вводить повідомлення коментаря в поле «Коментар». Та натискає «Надіслати»
Розширення сценаріїв	4.1 Система виявляє, що коментар занадто довгий 4.1.a Система демонструє повідомлення про максимальну довжину коментаря.

Таблиця 1.6 – Варіант використання UC006

Назва	Вхід до спільноти
Опис	Користувач може входити до існуючої спільноти
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та не належить даній спільноті
Постумови	Користувач належить даній спільноті

Продовження таблиці 1.6

Основний сценарій	Користувач входить на сторінку спільноти та натискає «Долучитись», система додає користувача до спільноти.
Розширення сценаріїв	

Таблиця 1.7 – Варіант використання UC007

Назва	Вихід користувача зі спільноти
Опис	Користувач може покинути спільноту за необхідністю
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та належить спільноті
Постумови	Користувач більше не належить спільноті
Основний сценарій	Користувач заходить на сторінку спільноти та натискає кнопку «Покинути спільноту». Система видаляє запис про належність користувача спільноті та направляє користувача на сторінку списку спільнот даного користувача.
Розширення сценаріїв	

Таблиця 1.8 – Варіант використання UC008

Назва	Голосування в опитуванні
Опис	Користувач може проголосувати в опитування в пості
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та належить спільноті, до якої належить пост з опитуванням
Постумови	Голос користувача врахувався в опитуванні
Основний сценарій	Користувач заходить на сторінку посту, голосує . Система запам'ятовує його голос
Розширення сценаріїв	

Таблиця 1.9 – Варіант використання UC009

Назва	Відміна голосу в опитуванні
Опис	Користувач може відмінити свій голос в опитуванні
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та належить спільноті, до якої належить пост з опитуванням, в якому користувач вже проголосував
Постумови	Голос користувача відмінився в опитуванні

Продовження таблиці 1.9

Основний сценарій	Користувач заходить на сторінку посту, бачить результати голосування в пості. Натискає кнопку «Відмінити голос»
Розширення сценаріїв	

Таблиця 1.10 – Варіант використання UC010

Назва	Зміна даних користувача
Опис	Користувач може змінити дані про себе на своїй сторінці
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі
Постумови	Дані користувача оновлені
Основний сценарій	Користувач заходить на власну сторінку профіля та натискає «Редагувати», вводить в відповідні поля свого профілю дані та натискає «Зберегти»
Розширення сценаріїв	

Таблиця 1.11 – Варіант використання UC011

Назва	Вихід з облікового запису
Опис	Користувач може вийти зі свого облікового запису

Продовження таблиці 1.11

Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та ввійшов в систему
Постумови	Користувач вийшов з системи
Основний сценарій	Користувач натискає на будь якій сторінці зверху кнопку «Вийти» та покидає обліковий запис.
Розширення сценаріїв	

Таблиця 1.12 – Варіант використання UC012

Назва	Користувач видаляє спільноту
Опис	Користувач може видаляти спільноти
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та є творцем спільноти
Постумови	Спільнота видалена
Основний сценарій	Користувач заходить даної спільноти та натискає кнопку «Видалити»
Розширення сценаріїв	

Таблиця 1.13 – Варіант використання UC013

Назва	Користувач видаляє коментар
Опис	Користувач може видаляти коментарі
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та є автором коментаря
Постумови	Коментар видалений
Основний сценарій	Користувач заходить на сторінку посту, де мав свій коментар. Натискає на кнопку «Видалити коментар»
Розширення сценаріїв	

Таблиця 1.14 – Варіант використання UC014

Назва	Голосування за пост
Опис	Користувач може голосувати за пост
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та входить в спільноту, до якої доданий пост
Постумови	Пост має голос користувача
Основний сценарій	Користувач заходить на сторінку з постом, натискає на кнопку «лайк» або «дизлайк». В системі оновлюються дані про голосування користувача за пост

Продовження таблиці 1.14

Розширення сценаріїв	
----------------------	--

Таблиця 1.15 – Варіант використання UC015

Назва	Користувач голосує за коментар
Опис	Користувач може голосувати за коментарі
Учасники	Користувач
Передумови	Користувач має обліковий запис в системі та належить спільноті, до якої доданий пост з даним коментарем
Постумови	Обновився голос за коментар
Основний сценарій	Користувач заходить на сторінку посту, та натискає кнопку «лайк» або «дизлайк» до відповідного коментаря. Система оновлює голос користувача за коментар
Розширення сценаріїв	

Також система має такі функціональні вимоги:

Таблиця 1.16 – Опис функціональної вимоги REQ001

Номер	REQ001
Назва	Реєстрація користувачів в системі
Опис	Система надає можливість користувачу реєструватись в системі

Таблиця 1.17 – Опис функціональної вимоги REQ002

Номер	REQ002
Назва	Створення спільнот
Опис	Система надає можливість зареєстрованому користувачу створювати спільноти.

Таблиця 1.18 – Опис функціональної вимоги REQ003

Номер	REQ003
Назва	Створення постів та опитувань
Опис	Система надає можливість зареєстрованому користувачу створювати пости та додавати в них опитування в разі необхідності.

Таблиця 1.19 – Опис функціональної вимоги REQ004

Номер	REQ004
Назва	Коментування постів та коментарів
Опис	Система надає можливість користувачу коментувати пости та коментарі.

Таблиця 1.20 – Опис функціональної вимоги REQ005

Номер	REQ005
Назва	Голосування в постах
Опис	Система надає можливість користувачу голосувати в постах

Таблиця 1.21 – Опис функціональної вимоги REQ006

Номер	REQ006
Назва	Оцінювання постів та коментарів
Опис	Система відображає користувачу оцінювати пости та коментарі

Таблиця 1.22 – Опис функціональної вимоги REQ007

Номер	REQ007
Назва	Оновлення даних сторінки
Опис	Система дозволяє користувачу оновлювати дані своєї сторінки

Таблиця 1.23 – Опис функціональної вимоги REQ008

Номер	REQ008
Назва	Видалення постів та коментарі
Опис	Система дозволяє користувачу видаляти свої пости та коментарі

Таблиця 1.24 – Опис функціональної вимоги REQ009

Номер	REQ009
Назва	Вихід та вхід зі спільноти
Опис	Система дозволяє користувачу вийти зі спільноти або увійти до неї

Таблиця 1.25 – Опис функціональної вимоги REQ010

Номер	REQ010
Назва	Видалення спільноти
Опис	Система дозволяє користувачу видаляти спільноти, які він створив

Таблиця 1.26 – Опис функціональної вимоги REQ011

Номер	REQ011
Назва	Вхід в обліковий запис
Опис	Система дозволяє користувачу входити в власний обліковий запис

Таблиця 1.27 – Опис функціональної вимоги REQ012

Номер	REQ012
Назва	Вихід з облікового запису
Опис	Система дозволяє користувачу вийти з облікового запису

Побудуємо результуючу матрицю трасування на Рисунок

	UC001 Авторизація користувача	UC002 Реєстрація користувача	UC003 Створення нової спільноти користувач	UC004 Створення посту	UC005 Коментування посту, коментарю	UC006 Вхід до спільноти	UC007 Вихід користувача зі спільноти	UC008 Голосування в опитуванні	UC009 Відміна голосу в опитуванні	UC010 Зміна даних користувача	UC011 Вихід з облікового запису	UC012 Користувач видаляє спільноту	UC013 Користувач видаляє коментар	UC014 Голосування за пост	UC015 Користувач голосує за коментар
REQ001 Реєстрація користувачів в системі		■													
REQ002 Створення спільнот			■												
REQ003 Створення постів та опитувань				■											
REQ004 Коментування постів та коментарів					■										
REQ005 Голосування в постах								■	■						
REQ006 Оцінювання постів та коментарів								■	■						■
REQ007 Оновлення даних сторінки										■					
REQ008 Видалення постів та коментарів													■		
REQ009 Вихід та вхід зі спільноти						■	■								
REQ010 Видалення спільноти												■			
REQ011 Вхід в обліковий запис	■														
REQ012 Вихід з облікового запису										■					

Рисунок 1.1 – Матриця трасування

1.5 Розробка нефункціональних вимог

Розроблюваний сервіс має відповідати наступним функціональним вимогам:

- відображення в браузерях Google Chrome версії 47.0 і вище, Firefox версії 54.0 і вище;
- передача даних по захищеному протоколу TLS;
- адаптивний інтерфейс.

1.6 Постановка задачі

Основною задачею створюваного сервісу є сприяння у проведенні плідних дискусій, голосувань, подальшому вивченню матеріалу читачами сайту.

Мета розробки – застосування теоретичних, практичних навичок, здобутих під час навчання для розробки WEB-сервісу для проведення онлайн дискусій та голосувань.

Призначення результату розробки – запуск проекту як бізнес-проекту для проведення онлайн дискусій та опитувань за допомогою створюваного проекту.

Задачі:

- авторизація користувачів у сервісі;
- створення спільнот;
- створення постів;
- коментування постів;
- оцінка постів та коментарів;
- створення опитувань;
- голосування та перегляд результатів опитувань.

Цілі:

- надати користувачеві інтерфейс для зручного проведення онлайн дискусій та голосувань;
- надати читачеві сайту сервіс зі зручним упорядкуванням інформації.

1.7 Висновки по розділу

У цьому розділі було описано та проаналізовано предметну область розробки, зокрема визначено критерії притаманні сервісам з подібною комунікаційною тематикою та необхідні цілі, яких має досягти система для задоволення поставлених завдань, розглянуто детально модель сервісу.

Було виділено успішні ІТ-проекти у даній області та виконано порівняння даного проекту с готовими продуктами, виходячи з порівняння можна зробити висновок, що додаток, розроблюваний в рамках цього дипломного проекту дещо відрізняється від інших тим, що зосереджується на тематичний та рейтинговий поділ у спільнотах, дає можливість голосувати та

					КПІ.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

має загальну тематику, це робить його роботу зосередженою для спільнот, що активно обговорюють та вирішують певні питання, в тому числі й питання саморганізації. Проте сервіс підійде і для інформаційно-розважальних цілей.

WEB-застосування повинно мати підтримувати досить широкий діапазон браузерів, щоб бути доступною користувачам. Функціональні вимоги до сервісу включають всі необхідні моделі та аспекти роботи представлені в розділі, проведено зіставлення функціональних вимог та варіантів використання, проаналізовано критерії коректності створюваної моделі.

					КПІ.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для створення програмного забезпечення необхідно провести детальний аналіз та моделювання[8], для чого буде використано методологію створення діаграм BPMN[9] та буде відображено основні процеси використання проекту користувачем на рисунках 2.1-2.7.

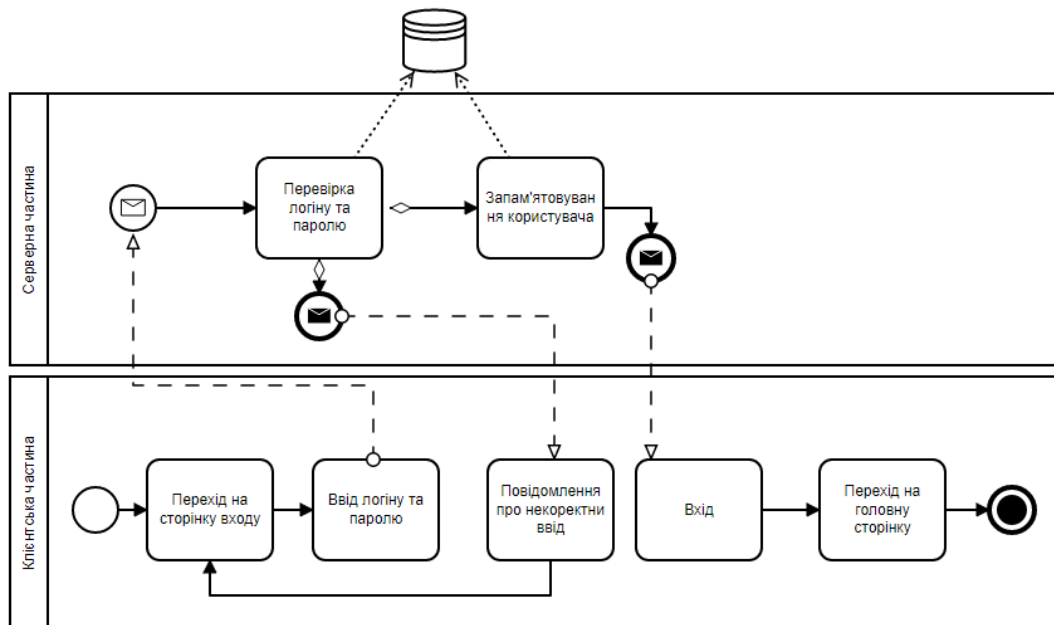


Рисунок 2.1 – Вхід користувача на сайт

Послідовний опис процесу входу користувача на сайт:

- користувач заходить на сторінку входу;
- вводить дані для входу;
- відбувається відправлення даних на сервер;
- якщо вони коректні – то сесія користувача запам'ятовується та повідомляється про успішний вхід;
- якщо некоректний, то повідомляється про помилку.

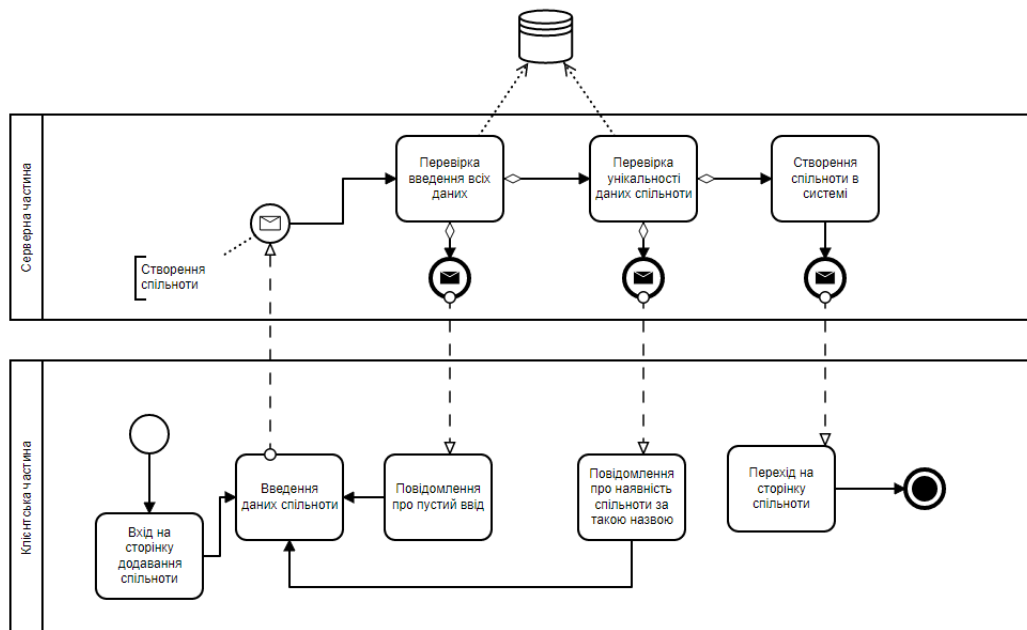


Рисунок 2.2 – Створення спільноти

Опис процесу створення спільноти:

- користувач входить на сторінку додавання спільноти;
- вводить дані нової спільноти;
- відправляє на сервер запит на створення;
- якщо все вірно, то сервер повідомляє користувача про успішне створення;
- інакше повідомляє користувача, що вже є наявна спільнота або назва є некоректною.

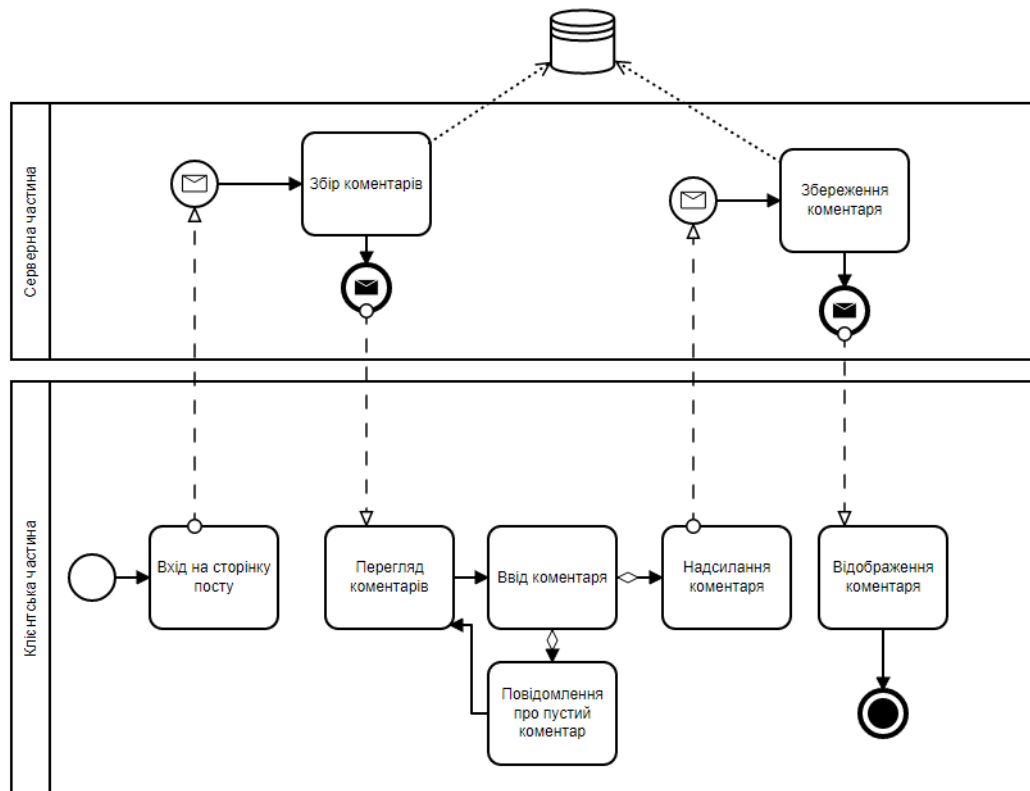


Рисунок 2.3 – Додання коментаря

Послідовний опис процесу додавання коментаря:

- вхід на сторінку посту;
- завантаження коментарів до посту;
- користувач шукає коментар до якого відповісти або відповідає до посту;
- вводить текст коментарю;
- відбувається відправлення запиту на сервер про створення коментаря;
- якщо дані коментаря коректні, то він зберігається в базу даних та відправляється користувачу як оновлений;
- інакше користувача повідомляють про помилку.

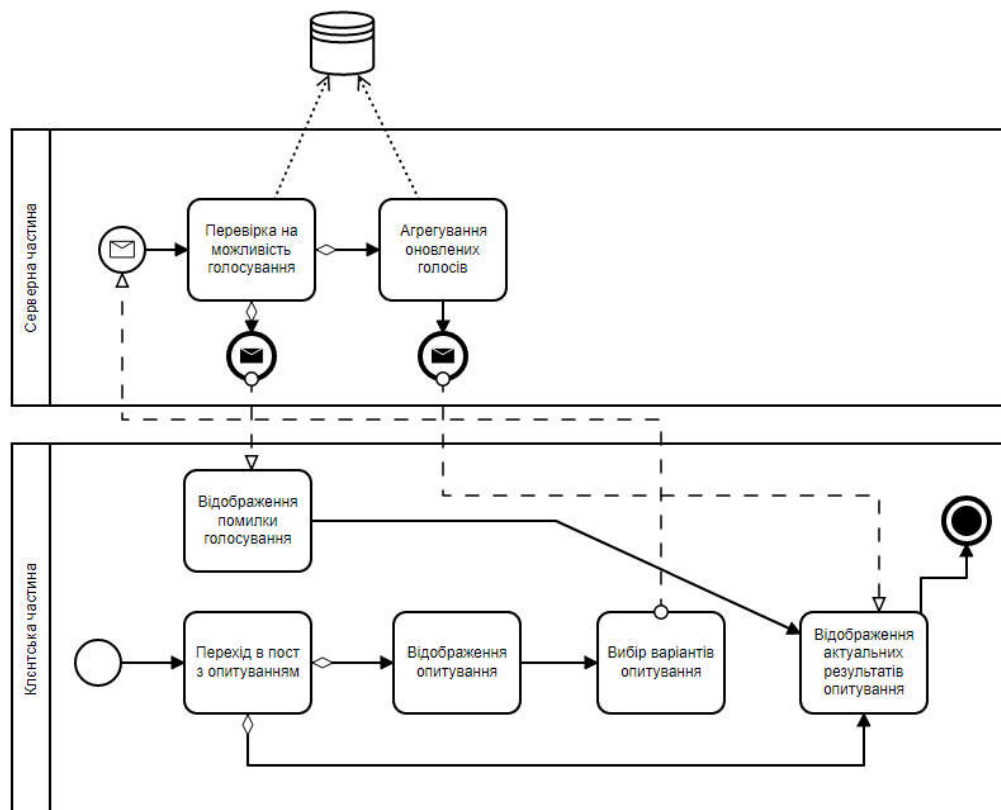


Рисунок 2.4 – Бізнес процес голосування

Опис бізнес процесу голосування:

- користувач заходить на пост з опитуванням;
- якщо він вже голосував, то отримає агреговані дані про голосування в опитуванні;
- якщо не голосував, то має список варіантів, за які він повинен голосувати;
- користувач голосує в опитуванні, дані відправляються на сервер;
- сервер перевіряє коректність даних, якщо вони коректні, то зберігає дані про голосування, агрегує нові та повертає користувачу;
- інакше повідомляє користувача про помилку.

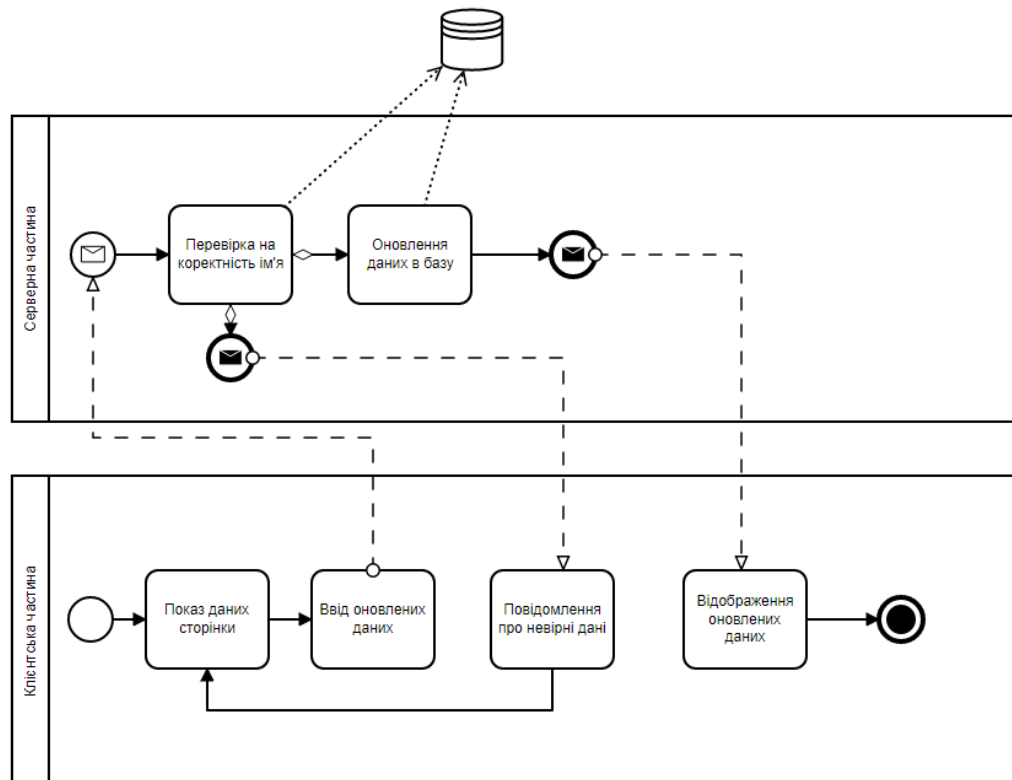


Рисунок 2.5 – Оновлення даних користувача

Послідовний опис процесу оновлення даних:

- користувач переходить на сторінку свого профілю;
- редагує дані та натиску кнопку оновлення;
- дані відправляються на сервер;
- якщо дані коректні, то вони оновлюються і повертається користувачу;
- інакше повідомляється про помилку оновлення.

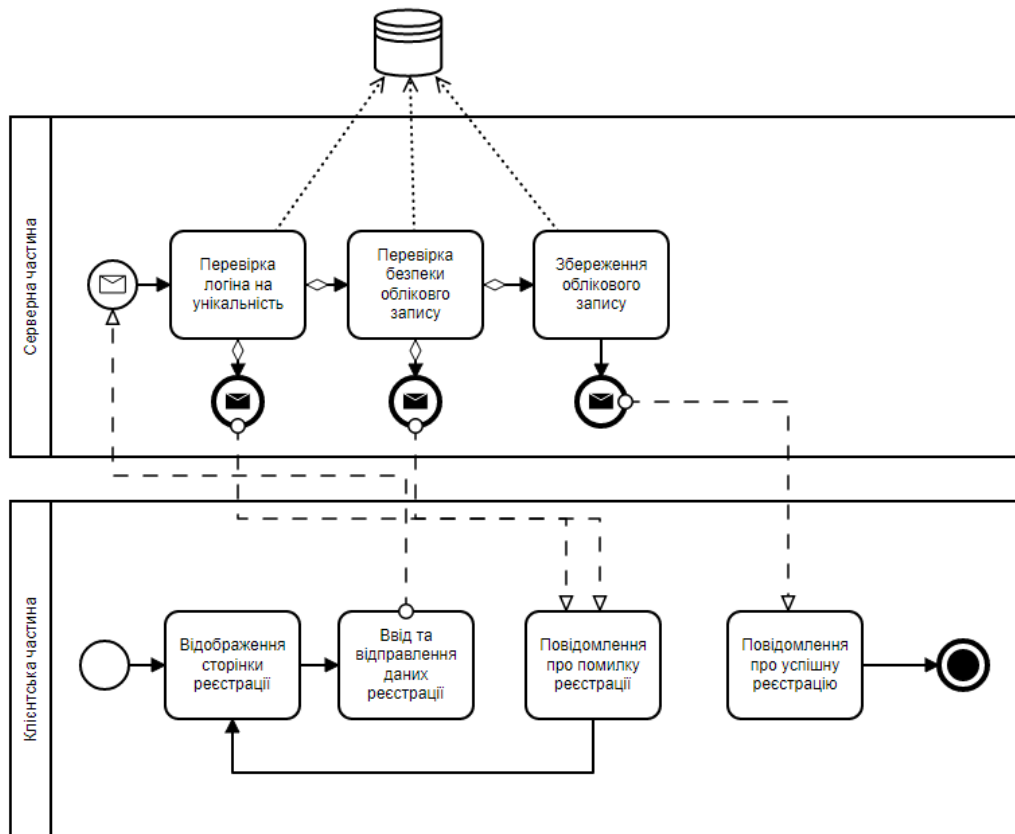


Рисунок 2.6 – Реєстрація користувача

Наведено послідовний опис процесу реєстрації користувача:

- користувачу відображується сторінка реєстрації;
- він вводить дані реєстрації;
- відправляється запит на реєстрацію на сервер;
- якщо дані не унікальні, то користувач повідомляється про некоректність даних;
- якщо дані не відповідають параметрам безпеки, то користувач повідомляється про невідповідність параметрам безпеки;
- якщо все вірно, то обліковий запис створюється, та користувач повідомляється про це.

2.2 Архітектура програмного забезпечення

Зобразимо загальну архітектуру клієнт-серверного застосування на Рисунок 2.7

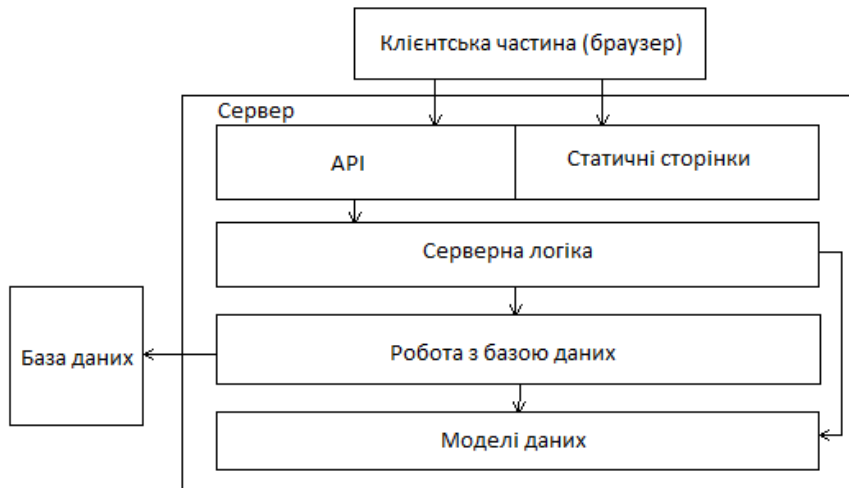


Рисунок 2.7 – Принципова архітектура дипломного проекту

Розроблена система буде виконуватись під операційними системами на базі ядра Linux, оскільки ця платформа є найвигіднішим та найзручнішим варіантом для проектів такого виду.

Використовується мова програмування F# [3] на платформі .NET Core[4] та фреймворку Giraffe[6] в якості серверної частини. Typescript[1] разом з бібліотекою React[7] для браузерної частини. PostgreSQL[5] в якості технології реляційної БД. ВЕБ-сервер працює за принципами SPA: сервер отримує надходження запиту, якщо це запит на сторінку повертає компільований Frontend-додаток, якщо запит на API – виконує необхідну обробку даних та повертає результат у вигляді JSON. Серверна частина складається з моделі, модулю роботи з базою даних, модулем логіки за стосунку та модулем будування pipeline-ів обробки запиту, засоби програмування на F# дає можливість лаконічно робити Dependency injection, побудову rout-ів, робити обробку даних.

Клієнтська частина складається з точки входу – яка в залежності від шляху запиту відображає потрібний компонент, компонентів - основних інтерфейсних блоків програми та міні-компонентів, що дозволяють повторне використання та використовуються в різних блоках, вони з себе представляють окремі елементи інтерфейсу.

2.3 Конструювання програмного забезпечення

Опишемо конструктивні складові серверної частини застосування

Таблиця 2.1 – Опис класів (структур) системи

Клас/модуль	Опис
Program	Вхідний клас програми, задає основну конфігурацію
AuthHandlers	Клас, що містить обробку авторизаційних запитів
EntityHandlers	Клас, що містить логіку обробки шаблонних запитів на об'єкти системи
Communities.Users	Модуль, що містить логіку обробку користувачів в спільнотах
BuildsListComponent	Клас, який містить бізнес-логіку компоненту списку побудов.
Communities.Posts	Модуль, що містить логіку обробки постів спільноти
Communities.Posts.Poll	Модуль, що містить логіку обробки опитувань
Users	Клас, що містить логіку обробки користувачів, збереження, валідацію даних
Auth	Клас, що містить логіку обробки авторизаційних даних

Продовження таблиці 2.1

Клас/модуль	Опис
ApiHandlers	Модуль, що містить допоміжну логіку обробки запитів на API частину
ApiHandlers	Модуль, що містить логіку обробки запитів на API
Models	Модуль, що містить основні структури даних проекту
DataAccess	Модуль, що містить всі запити до бази даних
DataAccess.EntityQueries	Клас, що містить логіку обробки шаблонних запитів на базові об'єкти системи
DataAccess.SqlQueries	Клас, що містить SQL-запити до бази даних
AuthUtil	Клас, що містить системну логіку роботи з авторизацією
Logger	Клас, що містить логіку для логювання
Common.Builders	Модуль, що містить основні елементи побудови виразів білдерів на F#
Settings	Клас, що містить обробку налаштувань додатку
Helpers	Клас, що містить допоміжні нерозподілені функції

Розглянемо детальніше кожен функцію по всім класам та функціям в таблиці, що буде наведена нижче

					КПІ.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Таблиця 2.2 – Опис методів класів та інтерфейсів системи

Клас/Модуль	Метод	Опис
Program	main()	Вхідна точка програми, налаштовує програму та сервіси згідно налаштувань
Program	configureLogging()	Налаштування логування даних
Program	configureServices()	Налаштування сервісів додатку
Program	configureApp()	Налаштування основної частини додатку
Program	cookieAuth()	Налаштування cookies
Program	errorHandler()	Обробка помилкових ситуацій
Logger	log(s: string)	Логування інформації
Logger	logError(s: string)	Логування помилки
Logger	logSql(sql: string)	Логування запиту на сервер бази даних
Logger	logWarn(s: string)	Логування за попередженням
Settings	getAppSettings()	Отримання налаштувань запуску
AuthHandlers	makeLogin()	Обробка запиту логіну в системі
AuthHandlers	makeRegister()	Обробка запиту реєстрації в системі

Продовження таблиці 2.2

Клас/Модуль	Метод	Опис
AuthHandlers	makeLogout()	Обробка запиту вихода користувача з облікового запису
EntityHandlers	getEntity()	Обробка запиту на базовий об'єкт системи
EntityHandlers	getEntities()	Змінює статус користувача на розробника або адміністратора.
Communities.Users	getIsUserInCommunity(userId: int, communityId: int)	Перевірити чи користувач є в спільноті
Communities.Users	addUserToCommunity(userId: int, communityId: int)	Додати користувача до спільноти
Communities.Users	findUserInCommunity(query: string, communityId: int)	Пошук користувача в спільноті
Communities.Users	removeUserFromCommunity(userId: int, communityId: int)	Видалення користувача зі спільноти
Communities.Posts	createPost(communityId: int, authodId: int, postModel: Post)	Створює пост
Communities.Posts	deletePost(postId: int)	Видаляє пост
Communities.Posts	getPostStatistics(postId: int)	Отримання статистики посту

Продовження таблиці 2.2

Клас/Модуль	Метод	Опис
Communities.Posts.Poll	createPoll(postId: int, data: PostData)	Створити опитування
Communities.Posts.Poll	voice(postId: int, voiceNumber, int)	Голосувати за пост
Communities.Posts.Poll	getStats(postId: int)	Отримати дані голосування
Users	getUserPageData(userId: int)	Отримати дані користувача для відображення на сторінці
Users	updateUserData(userId: int, data: UserData)	Обновити дані користувача
Auth	getHasExistingLogin(s: string)	Перевіряє чи є вже наявний користувач
Auth	registerUser(data: UserRegisterData)	Реєстрація користувача
Auth	logoutUser(sessionId: int)	Вихід користувача за сесії
Auth	loginUser(userId: int)	Вхід користувача в сесію
ApiHandlers	getCurrentUserHandler()	Отримати дані для заголовка в поточного користувача
ApiHandlers	searchOnSite()	Пошук на сайті

Продовження таблиці 2.2

Клас/Модуль	Метод	Опис
ApiHandlers	getUserFeed()	Отримати пости, які мають відображатись в стрічці новин користувача
ApiHandlers	getCommunityPageCardInfo()	Отримати деталі для сторінки спільноти
ApiHandlers	getCommunityPosts()	Отримати пости спільноти
ApiHandlers	getUserCardInfo()	Отримати деталі для карточки користувача
ApiHandlers	getUserPosts()	Отримати пости користувача
ApiHandlers	getPostData()	Отримати дані поста
ApiHandlers	getPostComments()	Отримати коментарі поста
ApiHandlers	getListCommunities()	Отримати список спільноти
ApiHandlers	getListUsers()	Отримати список користувачів
ApiHandlers	getPollInfo	Отримати дані опитування
ApiHandlers	createPost()	Створити пост
ApiHandlers	deletePost()	Видалити пост
ApiHandlers	votePost()	Оцінка поста

Продовження таблиці 2.2

Клас/Модуль	Метод	Опис
ApiHandlers	voteComment()	Проголосувати за коментар
ApiHandlers	votePoll()	Голосування в опитуванні
ApiHandlers	makeComment()	Створити коментар
ApiHandlers	enterCommunity()	Ввійти в спільноту
ApiHandlers	exitCommunity()	Вийти зі спільноти
ApiHandlers	createCommunity()	Створити спільноту
ApiHandlers	banUser()	Забанити користувача в спільноті
ApiHandlers	deleteCommunity()	Видалити спільноту
ApiHandlers	updateCommunity()	Оновити дані спільноти
Models	modelToJson(model: object)	Конвертувати модель в JSON
Models	jsonToModel(json: string)	Конвертувати JSON в модель
DataAccess	createTransaction()	Створити транзакцію до бази даних
DataAccess	createSqlQuery(sql: string, params: SqlParams)	Створити запит SQL
DataAccess	generateModelSqlQuery()	Згенерувати модель SQL

Продовження таблиці 2.2

Клас/Модуль	Метод	Опис
DataAccess.EntityQueries	getEntity(data: QueryData)	Отримати дані об'єкта
DataAccess.EntityQueries	getAllEntities(data: QueryData, paging: PagingData)	Отримати всі об'єкти за умовою
DataAccess.SqlQueries	getUserProfile()	Запит на профіль користувача
DataAccess.SqlQueries	searchInSite()	Пошук на сайті
DataAccess.SqlQueries	getUserFeed	Запит на стрічку новин користувача
DataAccess.SqlQueries	getCommunityPageCardInfo()	Отримати дані для сторінки спільноти
DataAccess.SqlQueries	getCommunityPosts()	Отримати пости спільноти
DataAccess.SqlQueries	getUserCardInfo()	Отримати дані для сторінки користувача
DataAccess.SqlQueries	getUserPosts()	Отримати пости користувача
DataAccess.SqlQueries	getPostData()	Отримати дані посту
DataAccess.SqlQueries	getPostComments()	Отримати коментарі посту
DataAccess.SqlQueries	getListCommunities()	Отримати список спільнот
DataAccess.SqlQueries	getListUsers()	Отримати список користувачів

Продовження таблиці 2.2

Клас/Модуль	Метод	Опис
DataAccess.SqlQueries	createPost()	Створити пост в спільноті
DataAccess.SqlQueries	deletePost()	Видалити пост зі спільноти
DataAccess.SqlQueries	votePost()	Оцінити пост значенням
DataAccess.SqlQueries	voteComment()	Оцінити коментар у пості
DataAccess.SqlQueries	makeComment()	Створити коментар до посту
DataAccess.SqlQueries	register	Зареєструватись в системі
DataAccess.SqlQueries	getLoginData()	Отримати інформацію про обліковий запис користувача
DataAccess.SqlQueries	checkCredentials()	Перевірити дані входу на правильність
DataAccess.SqlQueries	setTokenToUser()	Створити сесію користувача
AuthUtil	encryptPasswrod()	Створити шифрування для паролю, щоб зберігати його в базі даних
AuthUtil	generateToken()	Сгенерувати токен для сесії

Продовження таблиці 2.2

Клас/Модуль	Метод	Опис
Common.Builders	maybeBuilder	Створити білдер для Maybe монади
Common.Builders	resultBuilder	Створити білдер для Result монади
Common.Builders	asyncBuilder	Створити білдер для Async монади
Helpers	trimAll(s: string)	Видалити всі зайві пробіли

Розглянемо конструктивну структуру клієнтської частини додатку, вона складається за модулів логіки, віджетів та компонентів. Модулі логіки містять код для обробки даних, взаємодії з серверною частиною додатку. Віджети – це найголовніші інтерфейсні елементи додатку. Компоненти – допоміжні блоки для створення інтерфейсу. Перелік конструктивних елементів представлений в таблиці.

Таблиця 2.3 – Конструктивні елементи клієнтської частини

Назва	Тип	Опис
CommunityList	Віджет	Відображає список спільнот з фільтрацією і сортуванням
CommunityWidget	Віджет	Відображає сторінку спільноти, містить інформацію про спільноту та статистичні дані

Продовження таблиці 2.3

Назва	Тип	Опис
IndexWidget	Віджет	Представляє основний віджет застосунку, відображає найрелевантніші пости та основні операції
LoginWidget	Віджет	Відображення сторінки входу, взаємодія з сервером для входу
NewPostWidget	Віджет	Віджет для створення нового посту користувача
PostWidget	Віджет	Відображення посту та коментарів в ньому
RegisterWidget	Віджет	Сторінка реєстрації
UsersListWidget	Віджет	Віджет списку користувачів з сортуванням та фільтрами
UserWidget	Віджет	Відображення сторінки користувача
CommentTree	Компонент	Відображення дерева коментарів
CommunitySmallCard	Компонент	Картка спільноти в списках

Продовження таблиці 2.3

Назва	Тип	Опис
Header	Компонент	Шапка сайту з основними даними
OneComment	Компонент	Відображення коментаря
PostContent	Компонент	Відображення контенту посту
PostLong	Компонент	Відображення посту зі всіма даними
PostShort	Компонент	Preview посту в списках
TextEditor	Компонент	Редактор для вводу текстових даних
UserSmallCard	Компонент	Preview картки користувача в списках
CommonLogic	Модуль логіки	Логіка яка використовується в усьому клієнтському застосунку
CommonApi	Модуль логіки	Взаємодія з сервером
PollLogic	Модуль логіки	Логіка автооновлення даних опитування в пості
CommentLogic	Модуль логіки	Логіка автооновлення коментарів в пості

2.4 Аналіз безпеки даних

Безпека даних є багатогранною для такого застосунку, для забезпечення безпеки даних необхідно врахувати такі пункти:

- безпека зберігання даних в БД;
- безпека передачі даних;
- безпека обробки запитів до API;
- безпека роботи з сесіями користувача.

Для забезпечення безпеки зберігання даних в БД необхідно зберігати пароль користувача, зашифрувавши його за допомогою bcrypt[2], цей спосіб наразі вважається найбезпечнішим, оскільки це хешування має досить великий час роботи, що не дає його використовувати при підстановках паролю та надійний алгоритм хешування, який на даний момент ще не скомпроментований.

Безпека передачі даних передбачає шифрування виду Client-To-Server. Для цього буде застосовуватись стандарт HTTPS[10].

Безпека обробки запитів включає в себе екранування всіх строк для запитів до API, коректний набір операцій, що не дозволяє зробити запит з більш великим рівнем доступу.

Безпека роботи з сесіями реалізована через механізм від ASP.NET Core, що надає спосіб обробки запитів з обліковими даними.

2.5 Висновки по розділу

У даному розділі було створено архітектуру веб сервісу, обрано технології його розробки, зокрема: серверну мову програмування, що є зручною для такого класу задач, клієнтську мову програмування та фреймворк для створення інтерфейсу, також обрано реляційну СУБД, що є зручною, та має драйвер в серверній мові. Також було представлено структуру класів та функцій.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.4 Аналіз якості ПЗ

Якість програмного забезпечення – важлива характеристика продукту, оскільки вона є складовою актуальності для кінцевого користувача, зокрема є важливою для клієнт-серверних багатокористувацьких сервісів.

Для забезпечення якості одним із важливих процесів в розробці програмного забезпечення є його тестування. Тому воно є одним із головних етапів в при створенні програмних продуктів. Тестування дозволяє не допустити неякісний продукт до наступних етапів розгортання програмного продукту, завчасно знайти помилки в життєвому циклі розробки.

Для якісного тестування слід дотримуватись правил та стандартів[11], що були складені на основі досвіду в великій кількості програмних продуктів. Тому їх дотримання є необхідною умовою в процесах розробки. Зокрема існує такий стандарт тестування як IEEE 829—1998 Standard for Software Test Documentation[12].

До плану тестування увійде обсяг, підхід, ресурси та план усіх методів тестування. План описує програмні об'єкти що будуть протестовані, тип тестів, ресурси та план необхідний для виконання тестування.

У рамках цього плану буде виконано тестування частини продукту, що відповідає за основну логіку сайту.

У даному плані будуть протестовані наступні функції:

- реєстрація користувачів;
- авторизація користувачів;
- оновлення даних користувачів;
- створення спільнот;
- оновлення даних спільнот;
- вхід до спільнот;

					КПІ.ІП-5203.045440.02.81	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

- створення посту;
- коментування посту;
- оцінка посту та коментарю;
- створення опитування;
- голосування та перегляд результатів опитування.

Налаштовані наступні тестові модулі:

- реєстрація користувачів;
- авторизація користувачів;
- оновлення даних користувачів;
- створення спільнот;
- оновлення даних спільнот;
- вхід до спільнот;
- створення посту;
- коментування посту;
- оцінка посту та коментарю;
- створення опитування.
- голосування та перегляд результатів опитування.

3.5 Підходи до тестування

В рамках даного плану будуть використані наступні методи тестування:

- компонентне;
- інтеграційне;
- продуктивності.

3.5.3 Компонентне тестування

Методом компонентного тестування будуть перевірені логічно окремі частини веб сервісу, такі як:

- роути сервісу;

					КПІ.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

- методи доступу до бази даних;
- окремі допоміжні функції;
- методи авторизації.

3.5.4 Інтеграційне тестування

Методом інтеграційного тестування будуть перевірені взаємодії між модулями системи, такі як:

- взаємодія веб сервера з базою даних;
- взаємодія веб сервера з клієнтом.

3.5.5 Тестування продуктивності

Методом тестування продуктивності буде перевірена швидкодія наступних елементів системи:

- запити до бази даних;
- обробка сервером запитів без бази даних;
- обробка сервером запитів з базою даних.

3.6 Критерії проходження тестування

3.6.3 Компонентне тестування

Для компонентного тестування критерієм проходження є успішне виконання кожного пункту тесту. У разі якщо хоча б один пункт не був успішно виконаний – тестування вважається не пройденим.

3.6.4 Інтеграційне тестування

Для інтеграційного тестування критерієм проходження є успішне виконання кожного пункту тесту. У разі якщо хоча б один пункт не був успішно виконаний – тестування вважається не пройденим.

3.6.5 Тестування швидкодії

Для тестування швидкодії критерієм проходження є успішне виконання тесту з кожним доступним набором параметрів (кількість даних у одному запиті, кількість запитів, конкурентність) не довше ніж максимально допустимий час. У разі якщо хоча б один варіант тесту не був успішним або виконувався довше максимально допустимого часу – тестування вважається не пройденим.

3.7 Процес тестування

3.7.3 Дані до тестів

Вхідними даними для компонентного тестування є набори параметрів на яких очікується певний результат, що є вихідними даними даного тесту.

Вхідними даними для інтеграційного тестування є набори повідомлень що будуть передані від одного компоненту системи до іншого відповідно до конкретного тесту. Вихідними даними для даного виду тестування є результат роботи останнього компоненту у ланцюзі (наприклад, запис у базі даних у випадку тестування взаємодії API серверу та бази даних).

Вхідними даними до тестування швидкодії є набори даних, що покривають усі варіанти роботи системи у конкретному випадку. Вихідними даними є швидкість обробки запитів, кількість оброблених запитів, кількість неправильних реакцій на набір даних, дані по навантаженню на апаратну платформу (завантаженість процесору, вільна оперативна пам'ять, завантаженість мережі тощо).

3.7.4 Задачі тесту

Кожен тест повинен перевірити як правильність програми у відповідності до умов виконання тесту (test-driven development), так і виявити можливі помилки у роботі.

3.7.5 План виконання

Компонентне тестування повинне виконуватися до інтеграційного, яке, у свою чергу, виконується до тестування швидкодії.

3.8 Вимоги до середовища

3.8.3 Апаратна частина

Вимоги до апаратної частини співпадають з вимогами з технічного завдання.

3.8.4 Програмна частина

Для виконання тестування платформа повинна мати операційну систему на базі Linux.

3.8.5 Вимоги до безпеки

Для тестування бажано створити відповідний тестовий акаунт.

3.8.6 Інструменти

Для виконання тестування використовувати наступні програмні інструменти:

- Insomnia;
- Vegeta (<https://github.com/tsenart/vegeta>) – для тестування швидкодії.

3.9 Опис контрольного прикладу

Таблиця 3.1 – Перевірка входу на сайт

Мета тесту	Перевірка можливості входу
Початковий стан	Відкриті сторінка входу сайту
Вхідні дані	пара логін та пароль testAccount2 IncorrectPassword (не існуючий пароль);

Продовження таблиці 3.1

Схема проведення тесту	Ввід даних в поле та натиснення кнопки «Вхід»
Очікуваний результат	На сторінці відобразиться повідомлення про некоректний пароль

Таблиця 3.2 – Перевірка реєстрації на сайті

Мета тесту	Перевірити можливість зареєструватись на сайті
Початковий стан	Відкрита сторінка реєстрації
Вхідні дані	Повне ім'я: Юрій Ващенко, Стать: Чоловіча, Логін: Yura, Пароль, повторення паролю: qwerty
Схема проведення тесту	Ввід даних в поля «Повне ім'я», «Стать», «Логін», «Пароль», «Повторно пароль». Натиск на кнопку реєстрації
Очікуваний результат	Відбулась реєстрація, створився користувач з відповідними даними, перехід на сторінку користувача

Таблиця 3.3 – Створення спільноти

Мета тесту	Перевірити можливість створення спільноти
Початковий стан	Відкрита сторінка користувача, користувач увійшов в систему.
Вхідні дані	Назва спільноти, посилання на спільноту, яких ще не було в системі
Схема проведення тесту	Ввід даних в відповідні поля створення спільноти.
Очікуваний результат	Створилась спільнота з відповідними даними

Таблиця 3.4 – Створення посту

Мета тесту	Перевірити можливість створення посту
Початковий стан	Відкрита сторінка користувача, користувач увійшов в систему.
Вхідні дані	Заголовок посту та його текст
Схема проведення тесту	Ввід даних в відповідні поля створення посту.
Очікуваний результат	Був створений пост з відповідними даними

Таблиця 3.5 – Створення коментаря

Мета тесту	Перевірити можливість додати коментар на сайт
Початковий стан	Користувач увійшов в обліковий запис та відкрив сторінку з постом.
Вхідні дані	Обраний коментар або пост, до якого застосовується відповідь, коментар
Схема проведення тесту	Вибір об'єкта коментування, натиск на кнопку «Відповісти», ввід тексту коментаря, натиск на кнопку «Надіслати»
Очікуваний результат	Додався коментар

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.4 Розгортання програмного забезпечення

Для повного розгортання веб серверу необхідно:

- встановити основний сервіс;
- встановити СУБД PostgreSQL;
- налаштувати файл WebApp.dll.config;
- здійснити міграцію структури БД в СУБД;
- запустити основний сервіс.

4.4.3 Встановлення основного сервісу

Для встановлення основного сервісу необхідно взяти файли директорії /Bin/Server та помістити в необхідну робочу папку.

4.4.4 Встановити СУБД PostgreSQL

Для встановлення СУБД необхідно перейти на сайт PostgreSQL[4] та завантажити для необхідної платформи. Далі, користуючись інструкцією з встановлення встановити та вибрати порт, стандартного користувача та пароль. Необхідно, щоб доступ до серверу СУБД був з серверу з веб сервісом.

4.4.5 Налаштувати файл WebApp.dll.config

Необхідно налаштувати порт застосунку та дані доступу до бази даних, для цього слід заповнити поля файлу /Bin/WebApp.dll.config, а саме port, dbConnectionString

4.4.6 Здійснити міграцію структури БД в СУБД

Для здійснення міграції слід скопіювати вміст файлу /Sql/Initial.sql та виконати його в IDE для СУБД до потрібної бази даних.

4.4.7 Запустити основний сервіс

Для запуску серверу слід виконати команду `dotnet run WebApp.dll` до файлу, взятого з папки /Bin/WebApp.dll

4.5 Інструкція користувача

Повну інструкцію користувача наведено у документі «Керівництво користувача» дипломного проекту (КПІ.ІП-5203. 045440.05.34).

					КПІ.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

ВИСНОВКИ

В процесі розробки дипломного проекту був проведений повний цикл робіт для аналізу та створення програмного продукту. Він був розділений на складові, що є невід’ємними частинами проекту, для кожної був виділений розділ у дипломній документації.

У розділі “Аналіз вимог до програмного забезпечення” був проведений цілісний аналіз предметної області, вимог до програмного продукту та їх ролі в проекті.

У розділі “Моделювання та конструювання програмного забезпечення” було проведене моделювання застосунку, зокрема зроблені всі необхідні описові матеріали логічної складової сервісу, на основі цього було проведення конструювання, результатом якого було отримано цілісну архітектуру веб сервісу для проведення онлайн дискусій.

У розділі “Аналіз якості та тестування програмного забезпечення” було проаналізовано показники якості застосування, обґрунтовано необхідність тестування застосунку та наведено всі необхідні тестові матеріали.

У розділі “Впровадження та супровід програмного забезпечення” було описано процес впровадження сервісу, зокрема його встановлення на апаратну частину та належне налаштування, описано можливості для супроводу. Також було розроблено інструкцію користувача та адміністратора.

В рамках дипломного проекту були застосовані навички, отримані під час навчання, проаналізовано їх можливість та доречність в застосуванні відносно проекту та виведено необхідний набір матеріалу до проекту, зокрема були використанні навички побудови архітектури, засобів проектування, паралельного розподілення задач, захисту даних та мультипарадигменного програмування.

Розроблений дипломний проект є логічно цілісним та коректним, надійно протестованим, тому призначений для повноцінного застосування в цільових сферах комп’ютерних технологій.

					КПІ.ІП-5203.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Typescript Documentation [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.typescriptlang.org/docs/home.html>.
- 2) Bcrypt [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Bcrypt>.
- 3) F# Software Foundation [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://fsharp.org/>.
- 4) .NET Core Guide [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/core/>.
- 5) PostgreSQL [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/PostgreSQL>.
- 6) Giraffe [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://github.com/giraffe-fsharp/Giraffe>.
- 7) React Documentation [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://uk.reactjs.org>– React – Javascript бібліотека для створення користуачьких інтерфейсів.
- 8) Documentation in Software Architecture [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://medium.com/@nvashanin/documentation-in-software-architecture-4f2e4159c4fc>– стаття в мережі Інтернет.
- 9) BPMN Specification [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <http://www.bpmn.org/>.
- 10) HTTPS – Вікіпедія [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTTPS>.
- 11) Software Testing Methods [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <http://softwaretestingfundamentals.com/software-testing-methods/>.

12) IEEE 829 - 1998 Standart for Software Test Documentation
[Електронний ресурс]. – 2019. – Режим доступу до
ресурсу:<https://standards.ieee.org/standard/829-1998.html>.

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

***WEB-застосування проведення тематичних дискусій та on-line
опитувань***

Опис програми

КП.ІП-5203.045440.02. 13

“ПОГОДЖЕНО”

Керівник проекту:

_____ Ю.М. Крамар

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Ю.О. Ващенко

Київ – 2019 року

Тексти програмного коду**WEB-застосування проведення тематичних дискусій
та on-line опитувань**

(Найменування програми (документа))

CD-R

(Вид носія даних)

10 арк, 501 Кб

(Обсяг програми (документа) , арк.,) Кб)

S

Київ – 2019

					КПІ.ІП-5203.045440.02.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

Program.fs:
module App

open System

open System.Security.Claims

open System.Threading

open Microsoft.AspNetCore

open Microsoft.AspNetCore.Builder

open Microsoft.AspNetCore.Hosting

open Microsoft.AspNetCore.Http

open Microsoft.AspNetCore.Http.Features

open Microsoft.AspNetCore.Authentication

open Microsoft.AspNetCore.Authentication.Cookies

open Microsoft.Extensions.Configuration

open Microsoft.Extensions.Logging

open Microsoft.Extensions.DependencyInjection

open Giraffe

open System

open System.IO

open Giraffe.HttpHandlers

open Giraffe.Middleware

```
open Giraffe.Razor.Middleware
```

```
open AuthUtil
```

```
module Logger =
```

```
    let log = ignore
```

```
// Error handler
```

```
let errorHandler (ex : Exception) (logger : ILogger) =
```

```
    logger.LogError(EventId(), ex, "An unhandled exception occurred while  
executing the request.")
```

```
    clearResponse >=> setStatusCode 500 >=> text ex.Message
```

```
// config
```

```
let cookieAuth (o : CookieAuthenticationOptions) =
```

```
    do
```

```
        o.Cookie.HttpOnly    <- true
```

```
        o.Cookie.SecurePolicy <- CookieSecurePolicy.None
```

```
        o.SlidingExpiration   <- false
```

```
        o.ExpireTimeSpan <- TimeSpan.FromDays 7.0
```

```
let configureApp (app : IApplicationBuilder) =
```

```
    app.UseGiraffeErrorHandler errorHandler
```

```
    app.UseStaticFiles() |> ignore
```

```
    app.UseGiraffe Routes.webApp
```

```
    app.UseAuthentication() |> ignore
```

```
()
```

```
let configureServices (services : IServiceCollection) =
```

```
    let sp = services.BuildServiceProvider()
```

```
    let env = sp.GetService<IHostingEnvironment>()
```

```
    let viewsFolderPath = Path.Combine(env.ContentRootPath, "Views")
```

```
    services.AddRazorEngine viewsFolderPath |> ignore
```

```
    (services.AddAuthentication authScheme).AddCookie cookieAuth |> ignore
```

```
()
```

```
let configureLogging (builder : ILoggingBuilder) =
```

```
    let filter (l : LogLevel) = l.Equals LogLevel.Error
```

```
    builder.AddFilter(filter).AddConsole().AddDebug() |> ignore
```

```
[<EntryPoint>]
```

```
let main argv =
```

```
    let contentRoot = Directory.GetCurrentDirectory()
```

```
    let webRoot    = Path.Combine(contentRoot, "./public/build")
```

```
    WebHostBuilder()
```

```
        .UseKestrel()
```

```
        .UseContentRoot(contentRoot)
```

```
        .UseIISIntegration()
```

```
        .UseWebRoot(webRoot)
```

```
        .UseUrls("http://*:5000")
```

```
        .Configure(Action<IApplicationBuilder> configureApp)
```

```
        .ConfigureServices(configureServices)
```

```
        .ConfigureLogging(configureLogging)
```

```
        .Build()
```

```
        .Run()
```

```
0
```

```
PostLong.tsx:
```

```
import * as React from 'react';
```

```
import { withStyles } from '@material-ui/core/styles';
```

```
import Card from '@material-ui/core/Card';
```

```
import CardHeader from '@material-ui/core/CardHeader';
```

```
import CardMedia from '@material-ui/core/CardMedia';
```

```
import CardContent from '@material-ui/core/CardContent';
```

```
import CardActions from '@material-ui/core/CardActions';
```

```
import Collapse from '@material-ui/core/Collapse';
```

```
import { Avatar, Button } from '@material-ui/core';
```

```
import IconButton from '@material-ui/core/IconButton';
```

```
import Typography from '@material-ui/core/Typography';
```

```

import Link from '@material-ui/core/Link';
import red from '@material-ui/core/colors/red';

import FavoriteIcon from '@material-ui/icons/Favorite';
import CommentIcon from '@material-ui/icons/Comment';
import ThumbUpIcon from '@material-ui/icons/ThumbUp';
import ThumbDownIcon from '@material-ui/icons/ThumbDown';
import ShareIcon from '@material-ui/icons/Share';
import ExpandMoreIcon from '@material-ui/icons/ExpandMore';
import MoreVertIcon from '@material-ui/icons/MoreVert';
import classNames from 'classnames';
import api from '../commonApi'
import TextEditor from './TextEditor'

import PostContent from './PostContent'

const styles = theme => ({
  card: {
    width: '100%',
  },
  media: {
    height: 0,
    paddingTop: '56.25%', // 16:9
  },
  actions: {
    display: 'flex',
  },
  expand: {
    transform: 'rotate(0deg)',
    marginLeft: 'auto',
    transition: theme.transitions.create('transform', {
      duration: theme.transitions.duration.shortest,
    }),
  },
  expandOpen: {
    transform: 'rotate(180deg)',
  },
  avatar: {
    backgroundColor: red[500],
  },
});

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

class RecipeReviewCard extends React.Component<any, any> {
  state = Object.assign({}, this.props.model)

  makeLike = () => {
    api.votePost({
      postId: this.props.model.id,
      vote: 1
    }).then(x => {
      this.setState({likes: x.likes, dislikes: x.dislikes, myVote: x.user})
    })
  }

  makeDislike = () => {
    api.votePost({
      postId: this.props.model.id,
      vote: -1
    }).then(x => {
      this.setState({likes: x.likes, dislikes: x.dislikes, myVote: x.user})
    })
  }

  clickComment = () => {
    this.setState(state => ({commentExpanded: !state.commentExpanded}))
  }

  commentPush = (content) => {
    api.makeComment({
      parentCommentId: 0,
      content: content,
      postId: this.props.model.id
    }).then(x => {
      location.reload()
    })
  }

  render() {
    const { classes } = this.props;

    return (
      <Card className={classes.card}>

```

```

<CardHeader
  action={
    <IconButton>
      <MoreVertIcon />
    </IconButton>
  }
  title={this.props.model.title}
  subheader={new Date(this.props.model.time).toLocaleTimeString()}

/>

<CardContent>
  <PostContent model={this.props.model.content} />
</CardContent>
<CardActions className={classes.actions} disableActionSpacing>

  <IconButton onClick={this.makeLike}>
    <ThumbUpIcon />
    <Typography>
      {this.state.likes}
    </Typography>
  </IconButton>
  <IconButton onClick={this.makeDislike}>
    <ThumbDownIcon />
    <Typography>
      {this.state.dislikes}
    </Typography>
  </IconButton>

  <IconButton>
    <CommentIcon />
    <Typography>
      {this.props.model.countComments}
    </Typography>
  </IconButton>

  <IconButton aria-label="Share">
    <ShareIcon />
  </IconButton>

```



```

        <Typography><Link href={"/user/" +
this.props.model.author.url}>{this.props.model.author.name}</Link> in
community <Link href={"/community/" +
this.props.model.community.url}>{this.props.model.community.name}</Link></
Typography>
        <Button onClick={this.clickComment}>Comment</Button>
    </CardActions>

    <Collapse in={this.state.commentExpanded} timeout="auto"
    unmountOnExit>
        <CardContent>
            <TextEditor
            content="
            onSave={this.commentPush}
            onCancel={() => this.setState({commentExpanded: false})} />
        </CardContent>
    </Collapse>
</Card>
    );
}
}

```

```
export default withStyles(styles)(RecipeReviewCard);
```

UserSmallCard.tsx:

```

import * as React from 'react';
import { withStyles } from '@material-ui/core/styles';
import Card from '@material-ui/core/Card';
import CardHeader from '@material-ui/core/CardHeader';
import CardMedia from '@material-ui/core/CardMedia';
import CardContent from '@material-ui/core/CardContent';
import CardActions from '@material-ui/core/CardActions';
import Collapse from '@material-ui/core/Collapse';
import Avatar from '@material-ui/core/Avatar';
import IconButton from '@material-ui/core/IconButton';
import Typography from '@material-ui/core/Typography';
import Link from '@material-ui/core/Link';
import red from '@material-ui/core/colors/red';

```

```
import FavoriteIcon from '@material-ui/icons/Favorite';
```

```
import CommentIcon from '@material-ui/icons/Comment';
import ThumbUpIcon from '@material-ui/icons/ThumbUp';
import ThumbDownIcon from '@material-ui/icons/ThumbDown';
import ShareIcon from '@material-ui/icons/Share';
import ExpandMoreIcon from '@material-ui/icons/ExpandMore';
import MoreVertIcon from '@material-ui/icons/MoreVert';
import classNames from 'classnames';
```

```
import PostContent from './PostContent'
```

```
const styles = theme => ({
  card: {
    width: '90%',
  },
  media: {
    height: 0,
    paddingTop: '56.25%', // 16:9
  },
  actions: {
    display: 'flex',
  },
  expand: {
    transform: 'rotate(0deg)',
    marginLeft: 'auto',
    transition: theme.transitions.create('transform', {
      duration: theme.transitions.duration.shortest,
    }),
  },
  expandOpen: {
    transform: 'rotate(180deg)',
  },
  avatar: {
    backgroundColor: red[500],
  },
});
```

```
class RecipeReviewCard extends React.Component<any, any> {
  state = Object.assign({ expanded: false }, this.props.model)
```

```
render() {
  const { classes } = this.props;
```

```

return (
  <Card className={classes.card}>
    <CardHeader
      action={
        <IconButton>
          <MoreVertIcon />
        </IconButton>
      }
      title={this.props.model.name}
      onClick={() => {location.href = "/user/" + this.props.model.url}}
    />

    <CardContent>

    </CardContent>
    <CardActions className={classes.actions} disableActionSpacing>

      <IconButton>
        <ThumbUpIcon />
      </IconButton>

    </CardActions>
  </Card>
);
}
}

```

```
export default withStyles(styles)(RecipeReviewCard);
```

Змн.	Арк.	№ докум.	Підпис	Дата

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування проведення тематичних дискусій та on-line
опитувань

Технічне завдання

КП.ІП-5203.045440-03-91

“ПОГОДЖЕНО”

Керівник проекту:

_____ Ю.М. Крамар

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Ю.О. Ващенко

Київ – 2019 року

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
4.1	ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК.....	6
4.2	ВИМОГИ ДО НАДІЙНОСТІ	7
4.3	УМОВИ ЕКСПЛУАТАЦІЇ	7
4.4	ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ	7
4.5	ВИМОГИ ДО ІНФОРМАЦІЙНОЇ ТА ПРОГРАМНОЇ СУМІСНОСТІ	7
4.6	ВИМОГИ ДО МАРКУВАННЯ ТА ПАКУВАННЯ.....	8
4.7	ВИМОГИ ДО ТРАНСПОРТУВАННЯ ТА ЗБЕРІГАННЯ	8
4.8	СПЕЦІАЛЬНІ ВИМОГИ.....	8
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	9
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	10
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	11

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки:

Галузь застосування:

Наведене технічне завдання поширюється на розробку «WEB-застосування проведення тематичних дискусій та on-line опитувань » КПІ.ІП-5203.045440, котра використовується проведення онлайн дискусій та опитувань.

					КПІ.ІП-5203.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки «WEB-застосування проведення тематичних дискусій та on-line опитувань» є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім.Ігоря Сікорського).

					КПІ.ІП-5203.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для всіх людей, що цікавляться будь-якою соціальною, професійною чи науковою темою.

Метою розробки є покращення комунікаційних засобів для спілкування в онлайн-середовищі в дискусійному форматі та застосування опитування як засіб отримання наочної інформації щодо ставлення опитаних до того чи іншого явища.

					КПІ.ІП-5203.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1.1 Для користувача:

- створення акаунта;
- створення спільнот;
- створення постів;
- коментування постів;
- оцінка постів та коментарів;
- створення та голосування в опитуваннях.

4.2 Вимоги до надійності

4.2.1 Передбачити контроль введення інформації.

4.2.2 Передбачити захист від некоректних дій користувача.

4.2.3 Забезпечити цілісність інформації в базі даних.

4.3 Умови експлуатації

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.2 Обслуговування

4.3.3 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не пред'являються.

4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Мінімальна конфігурація технічних засобів:

процесор 1.1 ГГц і вище, 2 ядра;

– 1 Гб оперативної пам'яті;

– Інтернет-з'єднання швидкістю не менше 2Мбіт/с.

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем сімейства UNIX.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі:

Вхідні дані мають бути представлені у вигляді даних облікового запису користувача, його спільнот, статей, коментарів, оцінок, опитувань.

4.5.3 Результати повинні бути представлені в наступному форматі:

Вихідними даними є статті, коментарі, опитування, спільноти, що можуть бути розміщені в порядку цікавості, релевантності, а також оцінки, агреговані дані про кількість учасників та коментарів. Крім того вихідними даними є список користувачів.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги

Спеціальні вимоги відсутні.

					КПІ.ІП-5203.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 60 аркушах формату А4 (без додатків).

5.3.2 Опис програми

5.3.3 Технічне завдання

5.3.4 Програма та методика тестування

5.3.5 Керівництво користувача

5.4 Графічна частина повинна бути виконана на аркушах формату А3, котрі включаються у якості окремих документів:

5.4.1 Схема структурна варіантів використання

5.4.2 Схема бази даних

5.4.3 Креслення вигляду екранних форм

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

	Назва етапу	Строк	Звітність
1	Вивчення літератури за тематикою проекту	19.03.2019	
2	Розробка технічного завдання	26.03.2019	Технічне завдання
3	Аналіз вимог та уточнення специфікацій	15.04.2019	Специфікації програмного забезпечення
4	Проектування структури програмного забезпечення, проектування компонентів	22.04.2019	Схема структурна програмного забезпечення та специфікація компонентів
5	Програмна реалізація програмного забезпечення	29.04.2019	Тексти програмного забезпечення
6	Тестування програмного забезпечення	06.05.2019	Тести, результати тестування
7	Розробка матеріалів текстової частини проекту	13.04.2019	Пояснювальна записка.
8	Розробка матеріалів графічної частини проекту	20.05.2019	Графічний матеріал проекту
9	Оформлення технічної документації проекту	27.05.2019	Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-5203.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування проведення тематичних дискусій та on-line
опитувань

Програма та методика тестування

КП.ІП-5203. 045440.04.51

“ПОГОДЖЕНО”

Керівник проекту:

_____ Ю.М. Крамар

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Ю.О. Ващенко

Київ – 2019 року

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-ЗАСТОСУВАННЯ ПРОВЕДЕННЯ ТЕМАТИЧНИХ
ДИСКУСІЙ ТА ON-LINE ОПИТУВАНЬ

Програма та методика тестування

КПІ.ІП-5203.045440.04.51

“ПОГОДЖЕНО”

Керівник проекту:

_____ Ю.М. Крамар

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Ю.О. Ващенко

Київ – 2019 року

ЗМІСТ

1	ОБ'ЄКТ ВИПРОБУВАНЬ	3
2	МЕТА ТЕСТУВАННЯ	4
3	МЕТОДИ ТЕСТУВАННЯ	5
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ.....	6

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом тестових випробувань є веб сервіс для проведення онлайн дискусій та опитувань, створений як веб-сайт за допомогою фреймворків Giraffe, ASP.NET Core та React, з використанням PostgreSQL.

					КПІ.ІП-5203.045440.04.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 МЕТА ТЕСТУВАННЯ

Процес тестування полягав у перевірці програмних засобів за наступними критеріями:

- повна та цілісна інтуїтивна робота кожної сторінки веб-сервісу;
- швидка та якісна робота з сайтом;
- забезпечення надійного рівня захисту від некоректних запитів, що можуть призвести до порушення коректної роботи сервісу;
- належний та безпечний спосіб аутентифікації;
- відповідність програмних засобів до технічного завдання.

					КПІ.ІП-5203.045440.04.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 МЕТОДИ ТЕСТУВАННЯ

Тестування було виконане методом закритої коробки, таким чином перевіряючи як код, так і програмні засоби на повну відповідність функціональним та іншим вимогам до програмних засобів.

Використовуються наступні методи тестування:

- компонентне тестування;
- інтеграційне тестування;
- швидкодієнне тестування.

					КПІ.ІП-5203.045440.04.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується інструментаріями:

- Insomnia;
- Vegetta.

Надійність веб-ресурсу перевіряється методами:

- ручного тестування у відповідності до функціональних вимог;
- ручного тестування валідації;
- юніт-тестування коду;
- тестування зовнішнього вигляду та загального функціоналу у різних веб-переглядачах;
- тестування навантаженням;
- UX-тестування.

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування проведення тематичних дискусій та on-line
опитувань

Керівництво користувача

КП.ІІ-5203.045440.05.34

“ПОГОДЖЕНО”

Керівник проекту:

_____ Ю.М. Крамар

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ Ю.О. Ващенко

Київ – 2019 року

1 Інструкція користувача

Для користування веб сервісом необхідно перейти на його веб сторінку, при відкритті головної сторінки сайту користувач побачить список популярних постів на сайті, список популярних спільнот, також буде відображено кнопку входу та реєстрації.

Перш за все користувачу, що вперше на сайті необхідно зареєструватись, для цього треба перейти на сторінку реєстрації. Далі ввести своє повне ім'я, стать, логін та пароль. Пароль має бути не менш аніж 8 символів. В разі наявності поточно логіну на сайті, або закороткого паролю користувач отримає повідомлення про помилку. Сторінку входу зображено на рисунку 1.1

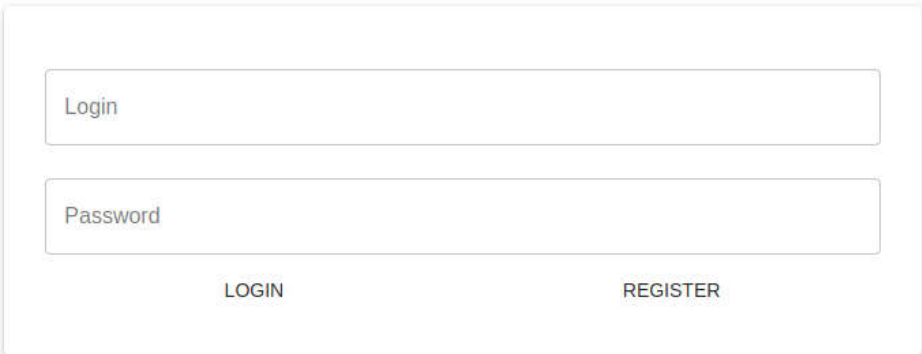


Рисунок 1.1 – Сторінка входу

При виборі входу користувач потрапляє на сторінку входу, де ввести свій логін та пароль, та натиснути на кнопку входу. В разі якщо дані входу некоректні, він отримає про це повідомлення.

Користувач, що увійшов, може також переглядати пости та коментарі, проте для створення постів, коментування та голосування слід увійти в відповідну спільноту. При вході в спільноту він отримає значення рейтингу в ній, що впливає на можливість коментувати та створювати пости. На рисунку 1.2 зображено коментарі в пості.

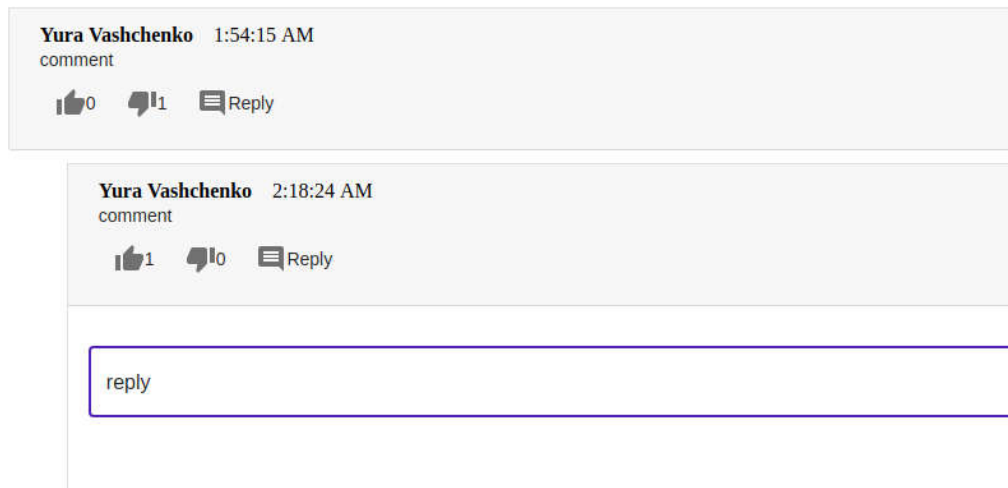


Рисунок 1.2 – Коментарі в пості

Для створення посту необхідно натиснути на кнопку «Створити пост» на своїй сторінці, або сторінці спільноти. В пості слід ввести його назву, текст посту, теги посту, також можна додати голосування.

Для коментування посту слід відкрити його сторінку, при цьому він має бути в тій же спільноті, що й користувач. Можна коментувати як пост, так і відповідати на коментар іншого учасника.

Користувач може змінювати дані своєї сторінки, для цього слід перейти на неї з шапки сайту та натиснути на кнопку «Редагувати», ввести в поля нові дані та натиснути кнопку зберегти.

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

WEB-застосування проведення тематичних дискусій та on-line
опитувань

Графічні матеріали

КПІ.ІП-5203.045440.06.99

“ПОГОДЖЕНО”

Керівник проекту:

_____ Ю.М. Крамар

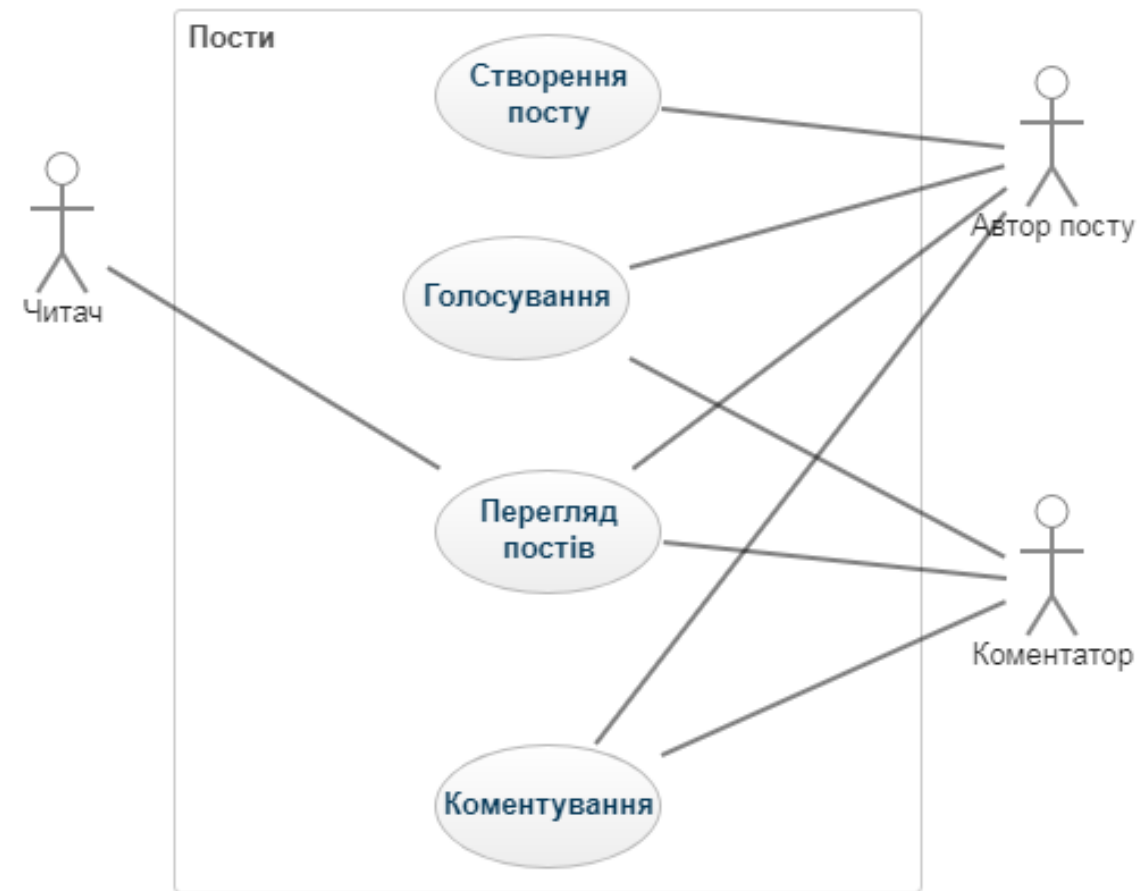
Нормоконтроль:

_____ К.І. Ліщук

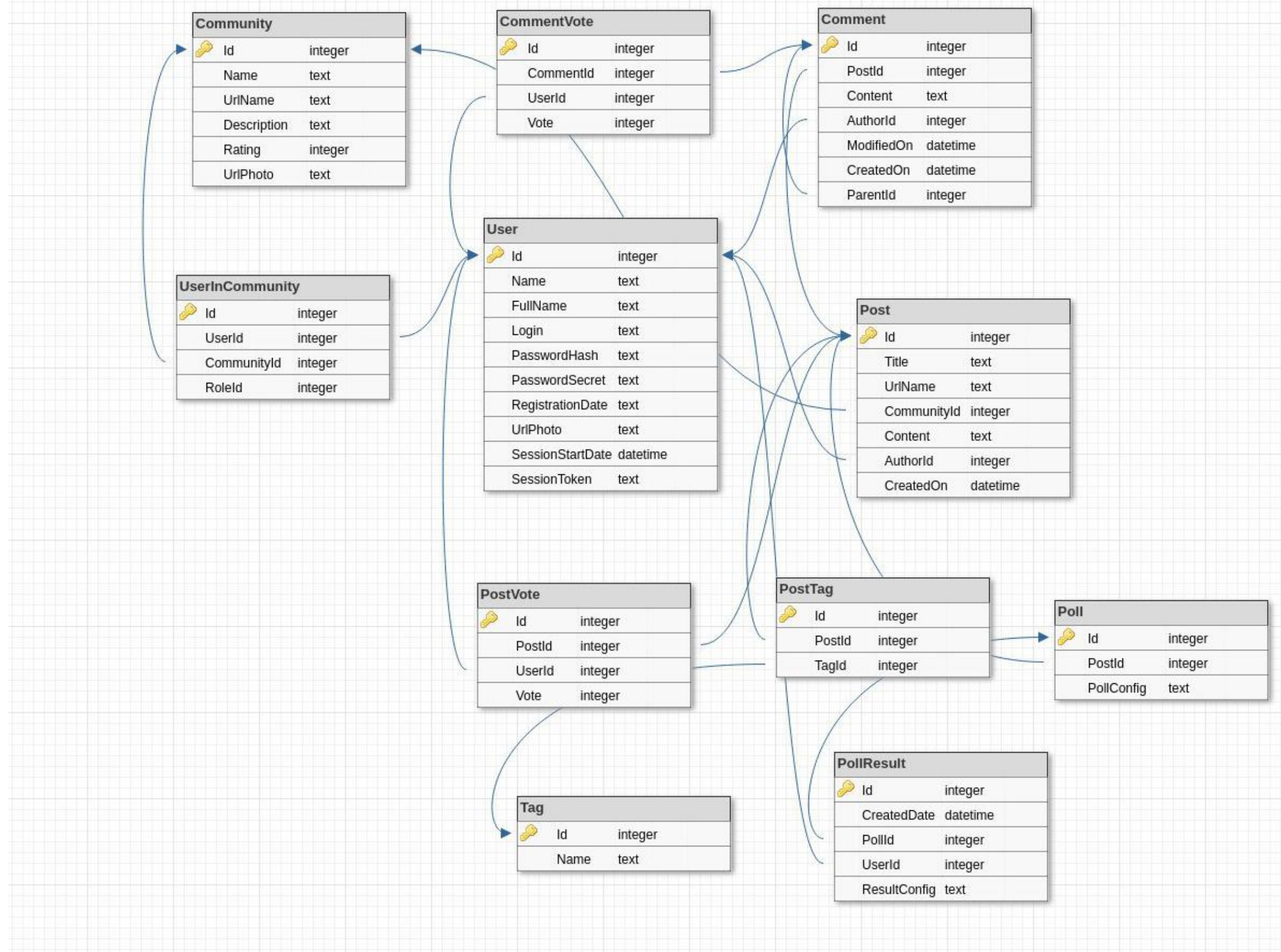
Виконавець:

_____ Ю.О.Ващенко

Київ – 2019 року



					КПІ.ІП-5203.045440.06.99.СС							
					Схема структурна варіантів використань	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив	Ващенко Ю.О.											
Перевірив	Крамар Ю.М.											
Т. кон.												
					WEB-застосування проведення тематичних дискусій та on-line опитувань	Аркуш		Аркушів				
Н. кон.	Ліщук К.І.					КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-52						
Затвердив	Крамар Ю.М.											



					КПІ.ІП-5203.045440.06.99 СБД						
					Схема бази даних						
Зм.	Арк.	№ документа	Підпис	Дата				Літера	Маса		Масштаб
Розробив		Ващенко Ю.О.									
Перевірив		Крамар Ю.М.									
Т. кон.								Аркуш		Аркушів	
					WEB-застосування проведення тематичних дискусій та on-line опитувань						
Н. кон.		Ліщук К.І.									
Затвердив		Крамар Ю.М.									
					КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-52						

Communism

Users 1

Posts 3

sadfsadf asdfs

1:24:25 AM

CONTENT

0 0 0 [Yura Vashchenko](#) in community [Communism](#)

asfs

1:24:15 AM

CONTENT

Yura Vashchenko 8:51:12 PM
fsfas

0 0 Reply

Yura Vashchenko 8:06:55 PM
sdfasdf sd fsdf sd fs df sdf s

1 0 Reply

Yura Vashchenko 7:49:43 PM
comment

0 1 Reply

Yura Vashchenko 8:46:25 PM
fsd

0 0 Reply

Yura

Name [Yura Vashchenko](#)
Communities

Pluses 1

Minuses 4

Posts

Comments 17

Yura Vashchenko 2:18:24 AM
comment

1 0 Reply

reply

SAVE

CANCEL

Login

Password

LOGIN

REGISTER

					КПІ.ІП-5203.045440.06.99 КЕ					
Зм.	Арк.	№ документа	Підпис	Дата	Креслення вигляду екранних форм	Літера			Маса	Масштаб
Розробив	Ващенко Ю.О.									
Перевірив	Крамар Ю.М.									
Т. кон.										
					WEB-застосування проведення тематичних дискусій та on-line опитувань	Аркуш			Аркушів	
Н. кон.	Ліщук К.І.					КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-52				
Затвердив	Крамар Ю.М.									